

CURRENT LISTING OF THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. -2. (Cancelled)

3. (Previously Presented) The method of claim 26, wherein determining the costs comprises determining the respective cardinalities of the first and second tables.

4. (Original) The method of claim 3, wherein determining the cost of applying the function on the second table comprises determining the cost of a join table that is a result of a join of the first table and another table.

5. (Original) The method of claim 3, wherein selecting the join path comprises applying the function on one of the first and second tables that has the lower cardinality.

6. (Original) The method of claim 5, wherein the function comprises a selection predicate applied on a complex attribute of the first table, the join query further containing a projection applied on a complex attribute of the first table, the method further comprising:
determining a cost associated with applying the projection on the first table and a cost associated with applying the projection on the join table,
wherein selecting the join path comprises applying the projection on one of the first table and the join table associated with a lower cost.

7. (Original) The method of claim 6, wherein selecting the join path comprises applying the projection on one of the first table and join table with the lower cardinality.

8. (Previously Presented) The method of claim 26, further comprising identifying the function as a costly function.

9. (Previously Presented) The method of claim 26, wherein the receiving, determining, and selecting acts are performed by the optimizer module.

10.-11. (Cancelled)

12. (Currently Amended) The article of claim 29[[11]], wherein the join query specifies the function being applied on the first table, and wherein the instructions when executed cause the database system to determine the join path in which the function is applied on the join table.

13. – 15. (Cancelled)

16. (Currently Amended) The article of claim 29[[11]], wherein the instructions when executed cause the system to determine the join path by further specifying a join of the join table and a third table to produce a fourth table.

17. (Previously Presented) The article of claim 16, wherein the join query further specifies application of a second function selected from the group consisting a selection predicate applied on a complex attribute, a projection applied on a complex attribute, and a user-defined data type method, the second function being applied on a third table, wherein the instructions when executed cause the database system to determine the join path by further applying the second function on one of the third table and the fourth table with a lower cardinality.

18. – 24. (Cancelled)

1 25. (Previously Presented) The method of claim 26, wherein the second table
2 contains a result of a join between the first table and another table,
3 wherein selecting the join path comprises selecting a join path in which at least
4 one of selection and projection is applied on the second table rather than the first table, the
5 method further comprising:
6 receiving a second query specifying a join of the first table and another table, the
7 second query specifying at least one of a selection predicate applied on a non-complex attribute
8 and a projection applied on a non-complex attribute; and
9 selecting another join path for the second query in which selection or projection is
10 applied on the first table before performing a join of the first table with the another table.

1 26. (Previously Presented) A method of performing a join in a database system
2 comprising:
3 receiving a join query containing at least one function selected from the group
4 consisting of a selection predicate applied on a complex attribute, a projection applied on a
5 complex attribute, and a user-defined data type method;
6 determining a cost associated with applying the function on a first table and a cost
7 associated with applying the function on a second table;
8 selecting a join path based on relative costs of applying the function on the first
9 and second tables; and
10 an optimizer module performing N-lookahead join planning in which costs for
11 different combinations of joins of N+2 tables are determined, where N is greater than or equal to
12 one.

1 27. (Cancelled)

28. (Currently Amended) The article of claim 29[11], wherein the instructions when executed cause the database system to:

receive a second query specifying a join of the first table and another table, the second query specifying at least one of a selection predicate applied on a non-complex attribute and a projection applied on a non-complex attribute; and

select another join path for the second query in which selection or projection is applied on one of the first and second tables before performing a join of the first table with the second table.

29. (Currently Amended) ~~The article of claim 11.~~ An article comprising at least one machine-readable storage medium containing instructions that when executed cause at least one processor of a database system to:

receive a join query containing at least one function selected from the group consisting of a selection predicate applied on a complex attribute, a projection applied on a complex attribute, and a user-defined data type method, the join query specifying a join of a first table and a second table to produce a join table;

determine a join path for the join query based at least in part on a cost associated with application of the function,

wherein determining the join path comprises selecting the join path in which the function is applied on the join table rather than the first table or second table to reduce cost; and

~~wherein the instructions when executed cause the database system to:~~

perform N-lookahead join planning in which costs for different combinations of joins of N+2 tables are determined, where N is greater than or equal to one.

30. (Cancelled)

1 31. (Previously Presented) A database system comprising:
2 a storage system to store tables; and
3 an optimizer to receive a join query that specifies a function selected from the
4 group consisting of a selection predicate applied on a complex attribute, a projection applied on a
5 complex attribute, and a user-defined data type method,
6 the optimizer to select a join plan based at least in part on a comparison of a first
7 cost of applying the function on a first table and a second cost of applying the function on a
8 second table,
9 wherein the optimizer is to perform N-lookahead join planning in which costs for
10 different combinations of joins of N+2 tables are determined, where N is greater than or equal to
11 one.

1 32. (Previously Presented) A method of performing a join in a database system,
2 comprising:
3 receiving a join query specifying a join of a first table and a second table and
4 containing at least one of a selection predicate and a projection;
5 selecting a join path for the join query in response to determining whether the at
6 least one of the selection predicate and projection is applied on a complex attribute,
7 wherein a first join path is selected in which the at least one of the selection
8 predicate and projection is applied on a join table instead of the first table and second table in
9 response to determining that the at least one of the selection predicate and projection is applied
10 on a complex attribute, the join table containing a join result of the first and second tables, and
11 wherein a second join path is selected in which the at least one of the selection
12 predicate and projection is applied on the first table before the join in response to determining
13 that the at least one of the selection predicate and the projection is applied on a non-complex
14 attribute.